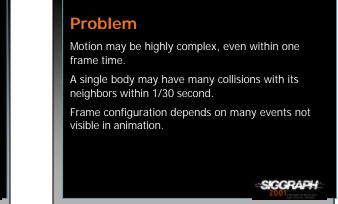


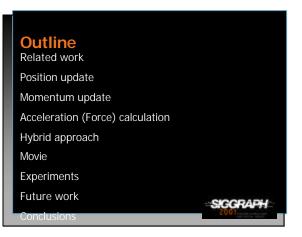


# Goal Animate many crowded rigid bodies.

SIGGRAPH



# Solution Generate plausible motion. Use iterated QP (quadratic programming) to generate positions/orientations, momenta, and forces at each frame time. Eye is "fooled" because brain cannot calculate complex inter-frame collisions. If still not realistic enough, hybridize with true animation to trade off running time and realism.



# Related Work (Plausible Animation): Position-Based Physics

Milenkovic SIGGRAPH 96

- +Can animate 1000 spheres in an "hourglass".
- +Requires only 1 minute/frame on 1GHz PC.
- -Zero-th order physics: no bouncing, friction, rotation, parabolic paths.
- -Acceleration has to be "faked".



SIGGRAPH

## Related Work (True Animation): Time-Warp

Mirtich SIGGRAPH 00

- +Desynchronizes collisions for bodies in different contact groups.
- +Up to N times faster than synchronized methods.
- -No desynchronization within one contact group.
- "Optimal" algorithm for "intractable" problem.



### **OBA Position Update: Target Positions/Orientations**

Bodies have linear and angular positions, velocities, and accelerations. Positions are updated in two-step process.

- 1. Calculate positions/orientations at next frame time under second order physics ignoring collisions
- --> target positions.
- 2. Calculate non-overlapping positions which minimize "distance" to target positions
- --> update positions.

#### OBA Position Update: Distance to Target Positions

Calculate (linear and angular) displacement from (proposed) UPDATE position to TARGET position

Plug displacement into formula for kinetic energy in place of velocity.

Heavy objects "push aside" light objects.

Also experimented with other positive-definite quadratic objectives.



## **OBA Position Update: Converting to Iterated QP**

Milenkovic 98: Rotational Compaction

Moves many rigid 2D polygons to non-overlapping positions/orientations which minimize a linear potential energy function.

Uses iterated LP (linear programming).

Generalize this algorithm to 3D polyhedra and a quadratic objective.



#### Rotational Compaction: A Few More Details

Decompose bodies into convex components.

Add separating plane for (some) pairs of convex components.

Add two additional variables per separating plane.

Add half-space constraints and linearize.

Solve resulting QP.

Iterate.



#### **Position Update Optimization**

Variables: linear and angular displacements (and separating plane orientations).

Objective: "distance" to target--plug displacements to target into formula for kinetic energy in place of velocities.

Constraints: bodies cannot overlap (linearized half-plane constraints).



#### **Collision and Static Contact** Response

Collisions: negative relative normal velocity; change body momenta with instantaneous impulses.

Static contacts: zero relative normal velocity; forces act over time.

Position update generates many simultaneous contacts: simultaneous QP models.

Friction is essential for physical realism.



#### **Momentum Update**

QP: simultaneous equal and opposite impulses at all colliding contacts (conservation of momenta).

Instantaneously change the body momenta.

Have a right-handed collision frame.

Impulses satisfy the empirical collision law.



Impulses push, but don't pull!

Impulses must satisfy the Coulomb friction law.

Introduce a friction cone: axis n, slope s = 1 / m

Confine impulse  $\vec{j}$  to the inside of the linearized



#### Implementation as a QP

Variables: body velocities and contact impulses.

Objective: total kinetic energy of all bodies after impulses (TD).

Constraints: impulses lie in linearized friction cones, and the collision law is satisfied.



#### **Acceleration (Force) Calculation**

Simultaneous QP model.

Reuse collision frame.

We can calculate the relative contact acceleration (Baraff, SIGGRAPH 89).

Note: there is a problem in our paper, constant velocity-dependent terms are missing.



#### **Frictionless Case (QP)**

Variables: body accelerations.

Objective: plug accelerations into sum of potential

and kinetic energies (SQP).

Constraint: non-negative contact normal

accelerations.

Note: the contact forces are implicit.



We verified empirically that this objective gives the exact force solution for the frictionless case.

Attempted to extend this to friction.



#### **Friction Case**

Introduce a friction-less acceleration cone constraint that mimics true friction.

This cone is perpendicular to the (dashed) force



#### Friction Case (QP)

Constraints: add to the frictionless QP the acceleration cone constraints.

Iterative update process can require excessive number of iterations.

We can impose a limit without visual instabilities.

Probably not better than standard methods: pivoting or penalty force methods.

We are working on something improved.



#### Friction Case (QP)

Constraints: add to the frictionless QP the acceleration cone constraints, and iterate.

Probably not better than standard methods: pivoting or penalty force methods.

We are working on something improved.



#### **Hybrid Scheme**

We generate target positions by ignoring collisions: these positions can be unrealistic.

Implement hybrid with retroactive detection (RD): allow limit of collisions for each pair, then ignore collisions.

Use heuristic to determine "bad" collisions.

Trade off speed for realism.



# Movie SIGGRAPH

#### **Experiments and Results**

Cost determined by QP solving.

Solving QP depends on number of constraints.

PSD: theoretical cost is polynomial of a high

Efficiency governed by position update (>50%): #constraints is proportional to #bodies *n*.

We achieve O( $n^2$ ) with CPLEX v7.0.



#### **Future Work**

- Better QP-based acceleration calculation.
- Other domains.
- Improve interaction with a priorianimation.
- Alternatives to the hybrid method.
- Parallel/distributed computation.



#### **Conclusions**

- · OBA allows efficient and stable simulation of large systems.
- It generates plausible motion.
- Uses readily available mathematical programming software.
- Bouncing and Newtonian trajectories.
- OBA can handle links and non-convexity.
- Hybrid approach to trade off speed for realism.
   SECFAPH

